# CDR-BASED CLOCK SYNTHESIS

## FIELD OF THE INVENTION

The invention relates generally to clock synthesis and, more particularly, to CDR (clock and data recovery)-based clock synthesis.

## BACKGROUND OF THE INVENTION

Conventional serial data transceivers are operable for transmitting and receiving serial data on a communication medium. Serial data transmission is controlled by a transmit clock signal. The transceiver device receives as an input a potentially noisy external clock source, and produces the transmit clock signal in response to this external clock source. The noisy external clock source is typically cleaned up by applying thereto a narrow-band filtering operation. Conventional approaches implement the narrow-band filtering with an analog implementation, which requires large capacitors. Thus, such analog implementations either occupy a large amount of die area within the transceiver, or must be implemented externally of the transceiver, thereby necessitating additional components and associated costs.

Generation of the transmit clock signal is also affected by a phenomenon known as frequency pulling. Frequency pulling exhibits itself as a low frequency modulation of the phase-locked loop (PLL) clocks and the transmit clock signal by the nearby receive clock signal that has been recovered from the incoming serial data, which receive clock signal has a frequency very close to the frequencies of the PLL clocks and the transmit clock signal. The modulation frequency can be determined by the ppm offset of the

respective clocks. The modulation amplitude depends on how and to what degree the two frequencies are coupled, e.g., through substrate and power supply. The problem of frequency pulling becomes more pronounced at higher levels of integration. In order to minimize coupling and crosstalk, some conventional approaches provide the receiver and transmitter as separate integrated circuits.

It is desirable in view of the foregoing to provide a serial data transceiver that can reduce frequency pulling and/or perform the aforementioned narrow-band filtering of an external clock source, while also maintaining acceptable levels of integration and cost.

Clock and Data Recovery (CDR) loops are used by conventional serial data transceivers to recover a clock from a received serial data stream and, subsequently, to recover and deserialize the received serial data. An example of such a conventional CDR loop, designated as CDR1, is illustrated diagrammatically in FIGURE 1. A reference clock signal REF_CLK is input to a PLL 11, which in turn generates quadrature I and Q clocks with a frequency to within parts per million (ppm) of the data rate of the serial data RXD_i received at input 13. The I and Q clocks are then duty cycle-corrected (DCC) and phase interpolated (PI). The phase interpolated signal ICLK at 14 is interpolated to be coincident to the data transitions, and the phase interpolated signal QCLK at 16 is interpolated to be centered in the middle of the data eye.

A bang-bang phase detector (PD) uses the clock signals ICLK and QCLK to provide at 15 phase error information which can be interpreted in the digital domain. Therefore, the succeeding stages in the CDR loop, for example the decimator 17, the loop filter (LPF) and the phase interpolator PI, can be implemented with digital or mixed-

signal techniques. The loop CDR1 outputs parallel data RD_i and the corresponding recovered clock.

**SUMMARY OF THE INVENTION**

A clock signal is synthesized by performing a CDR operation on a potentially noisy clock source signal which has a fixed transition density. The CDR operation produces a desired clock signal in response to the clock source signal. A single common PLL is used for both clock recovery and clock synthesis.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 diagrammatically illustrates a conventional example of a clock and data recovery circuit used at a serial data input of a conventional serial data transceiver.

FIGURE 2 diagrammatically illustrates pertinent portions of exemplary embodiments of a serial data transceiver apparatus according to the invention.

## DETAILED DESCRIPTION

A clock source can be thought of as a periodic data stream. Given a data rate of $f_D$, a clock source with a frequency of $f_D/(2 \times n)$ can be thought of as a periodic data stream with a (100/n) percent transition density. For example, clocks at frequencies of 1.25 GHz (n = 1), 625 MHz (n = 2), and 417 MHz (n = 3) are all 2.5 Gb/s periodic data streams with respective transition densities of 100%, 50% and 33%.

Performing CDR on a clock input is equivalent to recovering its frequency while attenuating its noise content. Unlike random data, a clock has a fixed (i.e., generally constant over time) transition density, so the CDR circuit can use a lower bandwidth than would be required for random data, thereby permitting rejection of phase noise at lower frequencies.

FIGURE 2 diagrammatically illustrates pertinent portions of exemplary embodiments of a serial data transceiver according to the invention. A noisy external clock source (TXCKSRC) can be applied (e.g., via selector 31) to an input 21 of the transceiver. The input 21 is coupled to the serial data input 13 of a CDR loop CDR2. In some embodiments, CDR2 can have the same structure and functionality as the loop CDR1 (see also FIGURE 1), although CDR2 can typically have a lower bandwidth than CDR1 because the input signal TXCKSRC has a known transition density, whereas CDR1 receives essentially random data RXD_i. The I and Q clock inputs of receive side loop CDR1 and transmit side loop CDR2 are produced by PLL 11 which is shared by

both loops CDR1 and CDR2 and is driven by the reference clock signal REF_CLK (see also FIGURE 1).

The transmit side loop CDR2 up-converts the incoming clock TXCKSRC to the serialization rate of the transmitted data, and also filters out noise content above the loop bandwidth, thereby minimizing jitter transfer. The transmit side loop CDR2 can thus generate cleaned-up clock signals, which can in turn be used in the transmit serialization process.

In some embodiments, the transmit clock signal produced by CDR2 in FIGURE 2 can be taken from the same phase interpolator output that produces ICLK in CDR1 (see also FIGURE 1), namely output 14 of CDR2 as shown in FIGURE 2. Other embodiments use phase interpolator output 16 (which produces QCLK in FIGURE 1) as the transmit clock signal. The transmit clock signal is used to serialize the data received from the transmit (TX) FIFO, resulting in output serial data TXD_i. If, for example, the clock source TXCKSRC has any of the three different frequencies specifically enumerated above (1.25 GHz, 625 MHz or 417 MHz), the transmit side loop CDR2 will produce the same transmit clock signal for any of the three input frequencies of TXCKSRC. Thus, the transmit serialization clock can be synthesized by performing a CDR operation on any one of a plurality of different frequency reference sources without requiring any frequency dividers. In some embodiments, the transmit clock frequency is equal to (serial transmit data rate)/2, so both the rising and falling edges of the transmit clock signal are used in the transmit serialization operation. The transmit clock signal is divided at 25 to produce a transmit byte clock signal TBC at a frequency equal to (serial

transmit data rate)/8. TBC serves as the clock source for the transmit parallel data stream which is to be serialized to 8 times the parallel data rate. Also some embodiments may require quarter or ½ data rate clocks for 4:1 and 2:1 serialization ratios.

Although the received data stream RXD_i and the transmit clock source TXCKSRC are, in general, plesiochronous (i.e., they have a frequency offset), the recovered receive clock RBC_i and the narrow-banded transmit clock at output 14 (or 16) of CDR2 are both generated by continuous phase interpolation of the synthesized clocks I and Q from the shared PLL 11. In this manner, the PLL clocks I and Q, the receive clock RBC_i, and the transmit clock have the same instantaneous frequency. The frequencies do, however, deviate relative to one another over longer observation periods.

The phase noise of the I and Q clocks from the PLL 11 is high-pass filtered (i.e., the low frequency noise content of the I and Q clocks is attenuated) by the loops CDR1 and CDR2, which leads to the attenuation of low-frequency phase modulations due to, for example, frequency pulling and flicker noise present at the outputs of PLL 11.

It will be apparent to workers in the art that inventive techniques described herein are applicable in transceivers for use with a variety of conventional data communication standards, for example: the SFI-5 standard promulgated by the Optical Internetworking Forum (OIF); the Xaui standard promulgated by the IEEE 802.3ae 10 Gigabit Ethernet Task Force; the FiberChannel standard promulgated by the X3T9.3 Task Group of ANSI; and the SONET standard promulgated by the Exchange Carriers Standards Association for ANSI.

In one example, the jitter bandwidth of the noisy clock source TXCKSRC (or the received serial data stream) is in the range of 1.5 MHz to 1.25 GHz for a 2.5 Gb/s data rate. In order to attenuate the clock jitter, some exemplary embodiments program the loop bandwidth of CDR2 lower than 1.5 MHz. Furthermore, for example, a 40 ppm offset between the frequencies of the receive clock (RBC_i) and the I and Q clocks produced by the PLL causes a modulation frequency of 100 KHz. Accordingly, in some exemplary embodiments, in order to eliminate the jitter due to frequency pulling, the loop bandwidth of CDR2 is programmed appreciably higher than 100 KHz. In general, if the clock source TXCKSRC is relatively noisy, the loop bandwidth should be reduced, and if TXCKSRC is relatively clean, then the loop bandwidth can be widened to reduce pulling effects.

Continuing with the foregoing example, if the loop bandwidth of CDR2 is between 100 KHz and 1.5 MHz, the noisy external clock is cleaned up, and jitter induced by frequency pulling is mitigated. For some exemplary embodiments, reducing phase noise of the common PLL 11 in the 5 KHz to 100 KHz range is critical (for example in SONET embodiments). Accordingly, in such embodiments, the loop bandwidth of CDR2 can be tuned towards the upper end of the aforementioned range, for example approximately 1 MHz. In other embodiments, reducing the incoming clock jitter in the MHz range and higher is important (for example in SFI-5 embodiments, in FiberChannel embodiments and in Xaui embodiments), so the loop bandwidth of CDR2 can be tuned toward the lower end of the aforementioned range, for example approximately 150 KHz.

FIGURE 2 also illustrates that, in some embodiments, either the external clock source TXCKSRC or the recovered receive clock signal RBC_i can be selectively applied to the input 21 by operation of a selector 31. The recovered clock is generally noisy and not suitable for data transmission without clean-up and filtering.

In some embodiments, as shown by broken line in FIGURE 2, a divide-by-two circuit is provided between selector 31 and CDR2 to reduce the effects of duty cycle distortion at input 13 of CDR2.

It should be clear from the foregoing that exemplary embodiments of the present invention utilize CDR techniques to narrow-band filter a noisy external clock source (or a recovered clock) in order to recover its frequency and remove its phase noise content above the CDR bandwidth. The CDR loop also attenuates the low frequency noise content of the clocks produced by the PLL below the loop bandwidth of the CDR, thus removing random VCO phase noise and phase modulation due to frequency pulling.

According to exemplary transceiver embodiments of the invention described above, the noisy external clock source is applied to the serial data input of a first clock and data recovery circuit. The first clock and data recovery circuit narrow-band filters the noisy external clock source in the digital domain, and produces the transmit clock signal. The first clock and data recovery circuit is driven by the same PLL clocks that drive a second clock and data recovery circuit used on the receive side to recover the incoming serial data. Both clock and data recovery circuits high-pass filter the phase noise of their shared PLL clocks, which reduces frequency pulling.

In some embodiments, a single common PLL is used for both clock recovery and clock synthesis in order to reduce the jitter caused by pulling between the asynchronous receive and transmit channels. Besides providing narrow-band filtering to the input clock source, the CDR attenuates low-frequency phase modulations of the PLL outputs.

Although exemplary embodiments of the invention are described above in detail, this does not limit the scope of the invention, which can be practiced in a variety of embodiments.